WHAT IS CLAIMED IS:

1. A method of dynamically verifying program operation, comprising:

executing a specified computer program;

while executing the specified computer program, maintaining a shadow array, the shadow array having entries corresponding to respective memory locations used by the specified computer program, each entry of the shadow array indicating a data type of the corresponding respective memory location;

the execution of the specified computer program including executing each of a plurality of instructions of the computer program, wherein execution of each instruction of a subset of the plurality of instructions includes:

determining whether execution of the instruction is inconsistent with an entry of the shadow array and generating a report when execution of the instruction is determined to be inconsistent with the entry of the shadow array;

executing the instruction; and

updating the shadow array in accordance with execution of the instruction.

2. The method of claim 1 wherein the execution of each instruction in the subset of the plurality of instructions includes:

identifying a memory location to be accessed by the instruction;

inspecting the shadow array entry corresponding to the identified memory location; and

determining whether execution of the instruction is inconsistent with the inspected shadow array entry.

3. The method of claim 2 wherein access of the memory location by the instruction comprises a read operation.

4. The method of claim 2 wherein access of the memory location by the instruction comprises a write operation.

5. The method of claim 1 wherein the determining includes determining whether proper execution of the instruction requires accessing data of a predefined data type that is different from the data type specified by the entry of the shadow array.

6. The method of claim 1 wherein the determining includes determining whether proper execution of the instruction is inconsistent with the data type specified by the entry of the shadow array.

7. The method of claim 1 wherein the respective memory locations include CPU registers, stack locations, and memory heap locations.

8. The method of claim 1 wherein the data type indicated by at least a subset of the shadow array entries indicates whether the corresponding memory location has been allocated.

9. The method of claim 1 wherein the data type indicated by at least a subset of the shadow array entries indicates whether the corresponding memory location has been initialized.

10. The method of claim 1, further comprising:
compiling a source code program into the specified computer program;
obtaining debugging information related to the specified computer program; and
initializing the shadow memory based on the debugging information.

11. The method of claim 1, further comprising not executing the instruction when execution of the instruction is determined to be inconsistent with the entry of the shadow array.

12. A computer program product for use in conjunction with a computer system, the computer program product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising:
a specified computer program;

a shadow array module for maintaining a shadow array, the shadow array having entries corresponding to respective memory locations used by the specified computer program, each entry of the shadow array indicating a data type of the corresponding respective memory location; and

an interpreter module for executing the specified computer program including executing each of a plurality of instructions of the specified computer program, wherein execution of each instruction of a subset of the plurality of instructions includes:

determining whether execution of the instruction is inconsistent with an entry of the shadow array and generating a report when execution of the instruction is determined to be inconsistent with the entry of the shadow array;

executing the instruction; and

updating the shadow array in accordance with execution of the instruction.

13. The computer program product of claim 12 wherein the execution of each instruction in the subset of the plurality of instructions includes:

identifying a memory location to be accessed by the instruction;

inspecting the shadow array entry corresponding to the identified memory location; and

determining whether execution of the instruction is inconsistent with the inspected shadow array entry.

14. The computer program product of claim 13 wherein access of the memory location by the instruction comprises a read operation.

15. The computer program product of claim 13 wherein access of the memory location by the instruction comprises a write operation.

16. The computer program product of claim 12 wherein the determining includes determining whether proper execution of the instruction requires accessing data of a predefined data type that is different from the data type specified by the entry of the shadow array.

17. The computer program product of claim 12 wherein the determining includes determining whether proper execution of the instruction is inconsistent with the data type specified by the entry of the shadow array.

18. The computer program product of claim 12 wherein the respective memory locations include CPU registers, stack locations, and memory heap locations.

19. The computer program product of claim 12 wherein the data type indicated by at least a subset of the shadow array entries indicates whether the corresponding memory location has been allocated.

20. The computer program product of claim 12 wherein the data type indicated by at least a subset of the shadow array entries indicates whether the corresponding memory location has been initialized.

21. The computer program product of claim 12, further comprising a compiling and debugging module for compiling a source code program into the specified computer program, and wherein the shadow array module further:

obtains debugging information related to the specified computer program from the compiling and debugging module; and

initializes the shadow memory based on the debugging information.

22. The computer program product of claim 12, further comprising not executing the instruction when execution of the instruction is determined to be inconsistent with the entry of the shadow array.

23. A computer program product for use in conjunction with a computer system, the computer program product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising:

a program instrumenting module for adding dynamic checking instructions to a compiled program to generate an instrumented program, the dynamic checking instructions including instructions for establishing a shadow array, the shadow array having entries

corresponding to respective memory locations used by the compiled program, each entry of the shadow array indicating a data type of the corresponding respective memory location;

the compiled program including a plurality of instructions;

wherein the dynamic checking instructions are configured so that during execution of instructions of the instrumented program, for each instruction of a subset of the plurality of instructions of the compiled program, the dynamic checking instructions:

determine whether execution of the instruction of the compiled program is inconsistent with an entry of the shadow array and generate a report when execution of the instruction is determined to be inconsistent with the entry of the shadow array; and

update the shadow array in accordance with execution of the instruction of the compiled program.

24. The computer program product of claim 23 wherein the determining comprises:

identifying a memory location to be accessed by the instruction;

inspecting the shadow array entry corresponding to the identified memory location; and

determining whether execution of the instruction is inconsistent with the inspected shadow array entry.

25. The computer program product of claim 24 wherein access of the memory location by the instruction comprises a read operation.

26. The computer program product of claim 24 wherein access of the memory location by the instruction comprises a write operation.

27. The computer program product of claim 23 wherein the determining includes determining whether proper execution of the instruction requires accessing data of a predefined data type that is different from the data type specified by the entry of the shadow array.

28. The computer program product of claim 23 wherein the determining includes determining whether proper execution of the instruction is inconsistent with the data type specified by the entry of the shadow array.

29. The computer program product of claim 23 wherein the respective memory locations include CPU registers, stack locations, and memory heap locations.

30. The computer program product of claim 23 wherein the data type indicated by at least a subset of the shadow array entries indicates whether the corresponding memory location has been allocated.

31. The computer program product of claim 23 wherein the data type indicated by at least a subset of the shadow array entries indicates whether the corresponding memory location has been initialized.

32. The computer program product of claim 23, further comprising a compiling and debugging module for compiling a source code program into the compiled program, and wherein the dynamic checking instructions further:

obtain debugging information related to the compiled program from the compiling and debugging module, and

initialize the shadow memory based on the debugging information.

33. The computer program product of claim 23, further comprising not executing the instruction when execution of the instruction is determined to be inconsistent with the entry of the shadow array.